

GULP.JS

Automate your front-end workflow
TJ Nicolaides - @tjnicolaides



WHAT IS GULP.JS?

- Command line task runner
- Built with Node.js
- Rich community of plugins
- Automatically manages front-end assets



WHY USE A BUILD TOOL?

- Write Less or Sass anywhere Node runs
- Watch directories to compress and concatenate JS and CSS, run tests, etc.
- Customize your project's development workflow
- Let Gulp do all the “Charlie work”



WHY NOT GRUNT?

- Gulp favors code over configuration - JS functions instead of nested config objects
- Gulp runs concurrent tasks out of the box
- Easier, generic API for writing plugins - it's just Node




```
tjnicolaides at Brownstone-024-MBPr in ~
$ npm install --global gulp

> v8flags@1.0.1 install /usr/local/lib/node_modules/gulp/node_modules/v8flags
> node fetch.js

flags for v8 3.14.5.9 cached.
/usr/local/bin/gulp -> /usr/local/lib/node_modules/gulp/bin/gulp.js
gulp@3.8.10 /usr/local/lib/node_modules/gulp
├─┬ interpret@0.3.7
├─┬ pretty-hrtime@0.2.2
├─┬ deprecated@0.0.1
├─┬ archy@1.0.0
├─┬ v8flags@1.0.1
├─┬ minimist@1.1.0
├─┬ semver@4.1.0
├─┬ tildify@1.0.0 (user-home@1.1.0)
├─┬ chalk@0.5.1 (escape-string-regexp@1.0.2, ansi-styles@1.1.0, supports-color@0.2.0, strip-ansi@0.3.0, has-ansi@0.1.0)
├─┬ orchestrator@0.3.7 (stream-consume@0.1.0, sequencify@0.0.7, end-of-stream@0.1.5)
├─┬ gulp-util@3.0.1 (lodash._reinterpolate@2.4.1, dateformat@1.0.8, vinyl@0.4.3, lodash.template@2.4.1, through2@0.6.3, multipipe@0.1.2, lodash@2.4.1)
├─┬ liftoff@0.13.6 (extend@1.3.0, flagged-respawn@0.3.1, resolve@1.0.0, findup-sync@0.1.3)
├─┬ vinyl-fs@0.3.11 (graceful-fs@3.0.4, strip-bom@1.0.0, mkdirp@0.5.0, vinyl@0.4.3, through2@0.6.3, glob-watcher@0.0.6, glob-stream@3.1.15, lodash@2.4.1)
tjnicolaides at Brownstone-024-MBPr in ~
$
```

INSTALLATION

<https://github.com/gulpjs/gulp/blob/master/docs/getting-started.md>

PROJECT SETUP

In our project root: gulpfile.js

```
var gulp = require('gulp');  
  
gulp.task('default', function() {  
  console.log('hello world');  
});
```


PROJECT SETUP

In our project root: package.json

```
{
  "name": "gulp-demo",
  "version": "1.0",
  "description": "A demo project showing off gulp.js",
  "main": "gulpfile.js",
  "repository": {
    "type": "git",
    "url": "https://github.com/tjnicolaides/gulp-demo.git"
  },
  "author": "TJ Nicolaides",
  "license": "ISC",
  "bugs": {
    "url": "https://github.com/tjnicolaides/gulp-demo/issues"
  },
  "homepage": "https://github.com/tjnicolaides/gulp-demo",
  "devDependencies": {
  }
}
```

PROJECT SETUP

```
gulp_project — ~/Sites/gulp_project — bash — 108x33

tjnicolaides at Brownstone-024-MBPr in ~/Sites/gulp_project on master [!?]
$ gulp
[22:51:51] Using gulpfile ~/Sites/gulp_project/gulpfile.js
[22:51:51] Starting 'default'...
hello world
[22:51:51] Finished 'default' after 63 µs

tjnicolaides at Brownstone-024-MBPr in ~/Sites/gulp_project on master [!?]
$
```


COMPILING SASS

Install our first dependency: gulp-ruby-sass

```
gulp_project — ~/Sites/gulp_project — bash — 108x33

tjnicolaides at Brownstone-024-MBPr in ~/Sites/gulp_project on master [!?]
$ npm install --save-dev gulp-ruby-sass
npm WARN package.json gulp-demo@1.1.0 No README data
gulp-ruby-sass@0.7.1 node_modules/gulp-ruby-sass
├── dargs@0.1.0
├── slash@0.1.3
├── each-async@0.1.3
├── win-spawn@2.0.0
├── chalk@0.5.1 (escape-string-regexp@1.0.2, ansi-styles@1.1.0, supports-color@0.2.0, strip-ansi@0.3.0, has-ansi@0.1.0)
├── glob@4.0.6 (inherits@2.0.1, graceful-fs@3.0.4, once@1.3.1, minimatch@1.0.0)
├── gulp-intermediate@3.0.1 (rimraf@2.2.8, uuid@1.4.2, mkdirp@0.5.0, through2@0.5.1, gulp-util@2.2.20)
├── gulp-util@3.0.1 (lodash._reinterpolate@2.4.1, dateformat@1.0.8, minimist@1.1.0, lodash@2.4.1, vinyl@0.4.5, through2@0.6.3, multipipe@0.1.2, lodash.template@2.4.1)
└─

tjnicolaides at Brownstone-024-MBPr in ~/Sites/gulp_project on master [!?]
$
```


COMPILING SASS

```
var gulp = require('gulp');
var sass = require('gulp-ruby-sass');

gulp.task('sass', function () {

    return gulp.src('scss/main.scss')
        .pipe(sass({sourcemap: true,
sourcemapPath: '../scss'}))
        .on('error', function (err)
{ console.log(err.message); })
        .pipe(gulp.dest('css'));

});
```


CLEANING DIRECTORIES

```
var del = require('del');  
  
gulp.task('clean', function (cb) {  
  del(['css/**'], cb);  
});
```


MANAGING JS ASSETS

```
var concat = require('gulp-concat');
var rename = require('gulp-rename');
var uglify = require('gulp-uglify');
var order = require("gulp-order");

gulp.task('minify', function(){
  return gulp.src('js/src/**/*.js')
    .pipe(order([
      'js/src/vendor/jquery-1.11.1.min.js',
      'js/src/**/*.js'
    ], { base: './' }))
    .pipe(concat('all.js'))
    .pipe(gulp.dest('js/dist'))
    .pipe(rename('all.min.js'))
    .pipe(uglify())
    .pipe(gulp.dest('js/dist'));
});
```


MANAGING JS ASSETS

```
var jshint = require('gulp-jshint');

gulp.task('lint', function() {
  return gulp.src(['js/src/**/*.js',
    '!js/src/vendor/*.js'])
    // lint all js files
    // but what's in the vendor dir
    .pipe(jshint())
    .pipe(jshint.reporter('default'));
});
```

OPTIMIZING IMAGES

```
var smushit = require('gulp-smushit');

gulp.task('smush', function () {
  return gulp.src('img/*.{jpg,png}')
    .pipe(smushit())
    .pipe(gulp.dest('img'));
});
```


WATCHING DIRECTORIES

```
gulp.task('default', ['sass'], function () {  
  gulp.watch("scss/**/*.scss", ['sass']);  
  gulp.watch('js/src/**/*.js', ['lint', 'minify']);  
});
```

GROUPING TASKS

```
gulp.task('build', ['clean', 'sass', 'smush',  
  'lint', 'minify']);
```


BROWSERSYNC

```
var browserSync = require('browser-sync');
var reload = browserSync.reload;

gulp.task('browser-sync', function() {
  browserSync({
    proxy: "gulp_project.dev"
  });
});

gulp.task('default', ['sass', 'browser-sync'], function () {
  gulp.watch("scss/**/*.scss", ['sass']);
  gulp.watch('js/src/**/*.js', ['lint', 'minify']);
});
```

AND MORE

- Run tasks on changed files only
- Compare filesizes before and after
- Inlining critical CSS
- Removing unused CSS
- Create sprites

EVEN MORE

- Auto-prefix CSS
- Crop and resize images, minify SVGs
- Compile Coffeescript
- Run Mocha or Karma tests
- Display Notification Center alerts

THANKS!

follow me - @tjnicolaides

